

This is a self-archived version of an original article. This version may differ from the original in pagination and typographic details.

Author(s): Sindhya, Karthik; Hakanen, Jussi

Title: Interactive Inverse Modeling Based Multiobjective Evolutionary Algorithm

Year: 2019

Version: Accepted version (Final draft)

Copyright: © Springer International Publishing AG 2019

Rights: In Copyright

Rights url: <http://rightsstatements.org/page/InC/1.0/?language=en>

Please cite the original version:

Sindhya, K., & Hakanen, J. (2019). Interactive Inverse Modeling Based Multiobjective Evolutionary Algorithm. In E. Minisci, M. Vasile, J. Periaux, N. R. Gauger, K. C. Giannakoglou, & D. Quagliarella (Eds.), *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences* (pp. 303-315). Springer. *Computational Methods in Applied Sciences*, 48. https://doi.org/10.1007/978-3-319-89988-6_18

Interactive Inverse Modeling Based Multiobjective Evolutionary Algorithm

Karthik Sindhya* and Jussi Hakanen

Abstract An interactive version of the inverse modeling based multiobjective evolutionary algorithm is presented. Instead of generating a representation of the whole Pareto optimal front, the algorithm aims at producing solutions in the regions where the decision maker is interested in. This is facilitated through an interactive solution process where the decision maker iteratively evaluates a set of solutions shown to her/him and the preference information obtained is used to adapt the search process of the algorithm.

1 Introduction

Recently, preference-based multiobjective evolutionary approaches have become an important research topic among evolutionary multiobjective optimization (EMO, see e.g. [4]). Instead of approximating the whole Pareto front, that is, the set of all Pareto optimal solutions in the objective space, it is typically of interest to approximate only some region of the Pareto front that the decision maker (DM) is interested in (see e.g. [1]). In order to enable the DM to learn about the behaviour of the problem considered and about his/her own preferences, (s)he should be able to change her/his preferences based on the solutions obtained during the solution process. This can be achieved by using interactive methods on top of preference-based multiobjective evolutionary approaches [8]. More information on interactive multiobjective optimization methods in general can be found in [10]. Interactive EMO algorithms are especially useful for problems having more than three objectives when the traditional EMO algorithms are no longer efficient [6].

Karthik Sindhya and Jussi Hakanen
Dept. of Mathematical Information Technology,
University of Jyväskylä, P.O. Box 35(agora), FI-40014 University of Jyväskylä, Finland
e-mail: (karthik.sindhya, jussi.hakanen)@jyu.fi

Several interactive multiobjective evolutionary algorithms have been proposed in the literature. For an extensive review of these interactive algorithms refer [11]. Here we present some prominent ones. In [5], a novel interactive optimization tool called I-EMO was proposed. Here first a set of Pareto optimal solutions were generated using an EMO algorithm and then a DM was helped through a graphical user interface to find her/his preferred solution(s) among them. Thiele et. al ([12]) proposed a preference based evolutionary algorithm for multiobjective optimization, wherein the DM progressively provided preference information in every generation as desirable objective function values and directed the search towards only preferable regions of the Pareto optimal front. In [6], Deb et al. proposed a progressively interactive EMO algorithm called PI-EMO. In PI-EMO algorithm, the DM is asked to provide preference information after every fixed interval of generations. The preference information is provided as a ranking of selected solutions shown in the current generation, which is subsequently used to construct an utility function. This utility function is further used to direct the search towards preferable solutions of the DM.

An emerging approach in EMO is estimation of distribution algorithms which instead of traditional reproduction operators build probabilistic models estimating the distribution of promising candidate solutions. New solutions are then produced by sampling the obtained distributions. A recent example of such an algorithm is the inverse modeling based multiobjective evolutionary algorithm (IM-MOEa) [3] which builds inverse Gaussian process based models from the objective space to the decision space in order to approximate the distribution of promising solutions. This enables generating new candidate solutions via sampling directly in the objective space.

This paper introduces an interactive IM-MOEa algorithm where the DM is able to guide the search towards her/his preferred regions of the Pareto front. To facilitate this, a decision making module is added to the IM-MOEa algorithm which takes into account the preferences of the DM for a prefixed number of interactions during the search. Through the decision making module, the DM is shown a number of solution candidates among which (s)he needs to select one, select two (that is, define a region in the Pareto front) or disregard all the solutions. Based on her/his actions, the search mechanism of the algorithm is adjusted accordingly.

The rest of the paper is divided into two parts. First, the proposed interactive IM-MOEa algorithm is described along with the necessary description of the original IM-MOEa algorithm. Then, the paper ends with a brief description of the numerical experiments and some concluding remarks.

2 Interactive IM-MOEA algorithm

2.1 IM-MOEA

The main idea of the IM-MOEA algorithm presented in [3] is to enable sampling in the objective space instead of the decision space. This is implemented by introducing Gaussian process based inverse mapping from the objective functions to the decision variables. When sampling in the objective space, the corresponding decision variable values are obtained by using the approximated inverse mapping. To ease the computation, the multivariate inverse model is decomposed into several univariate inverse models. The number of inverse models is reduced by using a random grouping strategy where some of the decision variables are assigned to a specific objective function. In practice, inverse modelling is performed inside sub populations created from the whole population with the help of fixed number of uniformly distributed reference vectors in the objective space. More information can be found in [3].

2.2 *Incorporation of decision maker preferences through adapting reference vectors*

The original IM-MOEA method [3] uses K uniformly distributed reference vectors to divide the population into sub populations. Each individual of the population is assigned to the closest reference vector and the solutions that are assigned to the same reference vector form a sub population resulting in K sub populations. In order to improve the performance for problems with discontinuous Pareto fronts, an adaptive reference vector scheme was proposed in [2]. In each generation, one of the reference vectors will be replaced by another randomly generated vector. In the beginning of the solution process, that is the exploration phase, the vector which has the most solutions assigned will be replaced. As opposed to that in the exploitation phase, the vector with the lowest number of solutions assigned will be replaced. More details of the adaptive version of the IM-MOEA algorithm can be found in [2].

The main idea of the interactive IM-MOEA algorithm presented here is to take advantage of the adaptive reference vector generation suggested in [2] and generate solutions that are desirable to the DM. Here, instead of replacing the reference vectors with randomly generated ones, such reference vectors will be added that can yield solutions reflecting the preferences expressed by the DM. The proposed interactive IM-MOEA algorithm consists of a decision making module attached to the original IM-MOEA algorithm. The interactive algorithm follows the general algorithm of IM-MOEA (see [3]) except the interaction step (step 4 shown in the Algorithm 1). The interaction step is executed not in each generation but only when interaction with the DM is needed. This can be determined, for example, based on how many times the DM wants to interact during the solution process (value

of this parameter can come from the decision). In that case, generations when the interaction is needed can be determined based on the number of interactions and the maximum number of function evaluations. Additionally, the crowding distance measure used for selection in [3] is replaced by a distance based selection in order to favour more solutions preferred by the DM. The details of the decision making module will be described in the next subsection.

Algorithm 1 The pseudo code of interactive IM-MOEA

```

1: initialization
2: while termination condition is not satisfied do
    3: partition of the combined population
    4: if interaction required then decision making module
    5: non-dominated sorting and selection
    6: for  $k = 1$  to  $K$  do
        7: inverse modeling
        8: reproduction
    9: end for
    10: update the combined population
11: end while

```

2.3 Decision making module

In this subsection we extend the IM-MOEA algorithm with the addition of a decision making module, which can be conveniently used to handle DM's preferences. As mentioned earlier, the decision making module presented in Algorithm 2 is based on the idea of adaptive reference vectors presented by Cheng et al. [2]. In addition to the parameters inherited from the original IM-MOEA algorithm, interactive IM-MOEA has two additional parameters, i.e. the number of interactions the DM wishes to have with the interactive IM-MOEA algorithm and the number of solutions (s)he wishes to see in each interaction. It must be noted that the DM has the freedom to change these two parameters during any interaction.

The decision making module is classified into four steps, i.e. classification of reference vectors, generate solutions for the DM, preference information from the DM and adjustment of reference vectors. In the first step, the current set of reference vectors are classified into active and inactive reference sets. The reference vectors with associated individuals form an active reference set and reference vectors with no associated individuals form an inactive reference set. In addition, the reference vector having the most solutions associated with is identified as the lead reference vector $\mathbf{v}^{\text{lead}}$. In the second step, the active reference set is clustered into N_{DM} clusters and one representative reference vector from each cluster is chosen. Next for each N_{DM} reference vectors, an individual that makes minimum angle with it is chosen. Thus N_{DM} individuals are shown as candidate solutions to the DM, based on which the DM provides her/his new preference information. The value for the parameter

Algorithm 2 Decision making module**Step 1:** *Classification of reference vectors:*

- For each reference vector $\mathbf{v}_i$, $i = 1, \dots, K$, calculate the number of individuals ($nsol_i$) of the population associated with it.
- The set of all reference vectors with no associated individuals form an inactive reference set IR and the rest of the reference vectors form an active reference set AR .
- From the set AR , the reference vector with maximum number of associated individuals is termed as lead reference vector $\mathbf{v}^{\text{lead}}$.

Step 2: *Generate candidate solutions for the DM*

- Cluster the elements of AR into N_{DM} clusters (If $|AR| < N_{DM}$, then $N_{DM} = |AR|$). From every cluster choose one reference vector as a representative index of the corresponding cluster. Subsequently, for every reference vector among all associated solutions, a solution $\mathbf{z}_i$ is chosen whose position in the objective space has minimum angle with the reference vector. Thus $\mathbf{z}_i$, $i = 1, \dots, N_{DM}$ candidate solutions are shown to the DM.

Step 3: *Preference information from the DM.* DM follows one of the following three paths:

- disregards all N_{DM} solutions shown to her/him and wishes to explore more. Go to *Step 4a*.
- chooses one solution among N_{DM} solutions as his preferred solution $\mathbf{z}_{\text{pref}}$. Go to *Step 4b*.
- identifies a preferred region defined by two solutions ($\mathbf{z}_{\text{pref}}$ and $\mathbf{z}_{\text{pref}+1}$) to explore further. Go to *Step 4c*.

Step 4: *Adjustment of reference vectors*

- *Step 4a:* Replace $\mathbf{v}^{\text{lead}}$ with a new randomly generated reference vector.
- *Step 4b:* The whole set of reference vectors is updated as follows: If this is not the last interaction, the extreme reference vectors $\mathbf{e}^i = (0, \dots, 1, \dots, 0)^T$, that is, unit vectors, are included. Then, the reference vector $\mathbf{v}^{\text{near}}$ having the minimum angle with the reference vector associated with $\mathbf{z}_{\text{pref}}$ ($\mathbf{v}^{\text{pref}}$) is added. Accordingly, the vector $\mathbf{v}^{\text{ref}}$ that is obtained by reflecting $\mathbf{v}^{\text{near}}$ with respect to $\mathbf{v}^{\text{pref}}$ is also added. The remaining reference vectors are uniformly distributed between $\mathbf{v}^{\text{near}}$ and $\mathbf{v}^{\text{ref}}$. The procedure is illustrated in Figure 1.
- *Step 4c:* The whole set of reference vectors is updated as follows: If this is not the last interaction, the extreme reference vectors $\mathbf{e}^i$ are included as in *Step 4b*. Then, the reference vectors associated with $\mathbf{z}_{\text{pref}}$ and $\mathbf{z}_{\text{pref}+1}$ are added and the rest of the reference vectors are equally distributed between these two vectors.

N_{DM} can be asked from the DM (that is, how many different solutions the DM wants to see at a time) or, otherwise, a default value can be used.

In step 3, the decision maker can either choose to explore further without choosing any solution, choose one solution or choose a preferred region identified by two different solutions. Based on what the decision maker chooses, different actions are taken in step 4. If the DM disregards all solutions shown to her/him in step 3, the lead reference vector $\mathbf{v}^{\text{lead}}$ is replaced by a randomly generated reference vector in *step 4a*. Alternatively, if the decision maker chooses one solution, all reference vectors other than the $\mathbf{v}^{\text{pref}}$ (i.e. the one that the preferred solution is associated with) are deleted. The entire set of reference vectors is regenerated between the reference vector $\mathbf{v}^{\text{near}}$ that is the closest with the $\mathbf{v}^{\text{pref}}$ and its reflected counterpart $\mathbf{v}^{\text{ref}}$ with respect to $\mathbf{v}^{\text{pref}}$ as shown in Figure 1. This is done in order to bias the search towards

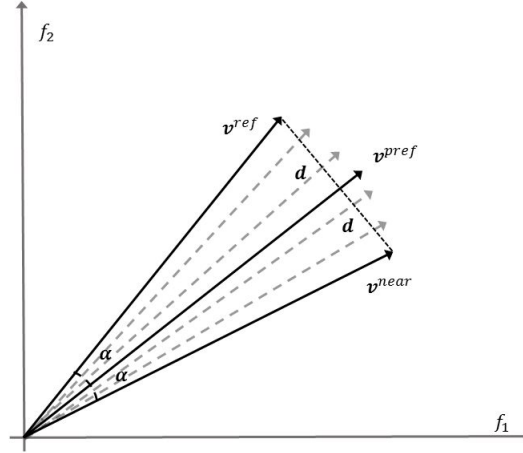


Fig. 1 Illustration of reference vector generation in step 4b of the Algorithm 2.

region of interest to the decision maker. Finally, if the decision maker prefers two solutions (i.e. is interested in the region between them), the set of reference vectors is generated between the two reference vectors that the preferred solutions indicated by the DM are associated with. Additionally, at each interaction (except the final), the extreme reference vectors are included to be able to allow changes in preference information and maintain diversity in the population. They are not included in the final interaction in order to guarantee convergence according to preferences specified by the DM. Note that the number of the reference vectors remains fixed throughout the search.

3 Numerical experiments

The performance of the interactive IM-MOEA algorithm consisting of the proposed decision making module in conjunction with the IM-MOEA algorithm is demonstrated here by using two and three objective problems. The problems selected are bound constrained problems F2 (two objective problem, modified from the ZDT2 problem) and F4 (three objective problem, modified from the DTLZ2 problem) from [3], both having 30 decision variables. The population size used was 50 for both problems and the maximum number of function evaluations was 50000 and 100000 for F2 and F4, respectively. The number of reference vectors was set to $K = 10$ for both problems as suggested in [3]. The random group size was set little higher ($L = 10$) than proposed in [3] in order to get more accurate results. Clustering in step 2 of the decision making module is performed by using K-means clustering [7]. The interaction between the DM and the interactive IM-MOEA algorithm for both prob-

lems F2 and F4 are summarized in Table 1. In every interaction the solutions chosen by the DM are marked in bold font.

3.1 Example with a biobjective problem

In the first example, problem F2 with a nonconvex Pareto front will be used. The DM wants to interact four times ($N_{ia} = 4$) and he wants to see five solutions at each interaction ($N_{DM} = 5$). Initially, his preferences are to get as close to $f_1 = 0.5$ as possible. In the first interaction, the candidate solutions shown to him are presented in Table 1. Note that there are only four candidate solutions since only four of the reference vectors are active (i.e., have associated solutions) at this phase.

The range of the solutions shown for f_1 is $[0.000, 0.415]$ and, according to his preferences, he selects candidate solution number 4 as the preferred one. Figure 2 shows the candidate solutions, a solution selected by the DM and the updated set of the reference vectors where the bounding reference vectors are highlighted. Note that it is only possible to show this kind of visualization to the DM when the number of objective functions is two or three. For problems with more objectives, other types of visualization techniques need to be used (see e.g. [9]). In addition, reference vectors are not meant to be shown to the DM but are in this paper included for illustrative purposes. Based on the preferences, the candidate solutions shown to the DM for the second interaction are presented in Table 1. This time, there are five candidate solutions since at least five reference vectors have associated solutions.

Candidate solution 1 has a value very close to the initial preferences of the DM ($f_1 = 0.507$). Next, he wants to see more solutions between candidate solutions 1 and 2 and selects them as preferred solutions. Figure 2 shows the candidate solutions, solutions selected by the DM and the updated set of the reference vectors where the bounding reference vectors are highlighted. Note that now the spread of the reference vectors is smaller when the DM wants to zoom in to a specific region in the Pareto front. Based on those preferences, the next five candidate solutions are shown in Table 1.

Now all the candidate solutions except number 1 are very close to the initial preferences of the DM. Having examined the solutions in more detail, he is not interested in them anymore but wants to improve f_1 further. To zoom out from the region of his initial preferences, he selects the candidate solutions 1 and 3 as he wants to examine the region for lower f_1 values. Accordingly, Figure 2 shows the candidate solutions, solutions selected by the DM and the updated set of the reference vectors where the bounding reference vectors are highlighted. Note that now the spread of the reference vectors is bigger since the DM wants to zoom out. The resulting candidate solutions are now shown in Table 1.

Based on the candidate solutions shown, the DM was able to move towards a region where f_1 values are smaller than 0.5. This is now the final interaction and he wants to focus in the area between candidate solutions 2 and 3. Figure 2 shows the candidate solutions, solutions selected by the DM and the updated set of the

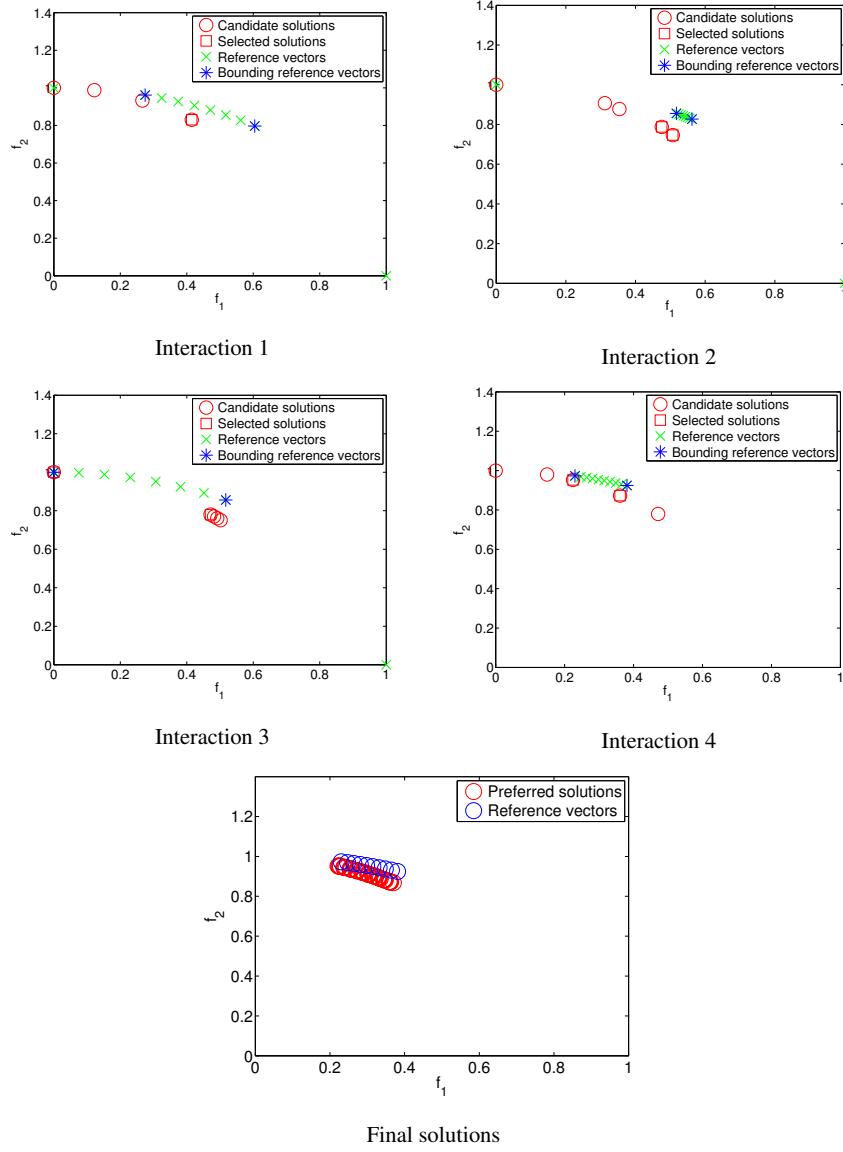


Fig. 2 Interactions with the DM for Problem F2

reference vectors where the bounding reference vectors are highlighted. Note that after the last interaction, the extreme reference vectors are not anymore included since the focus is now on converging according to the last preferences. The set of final solutions obtained is shown in Figure 2 along the final set of reference vectors. As can be seen, the solutions obey nicely to the final set of reference vectors

that are based on the last preferences of the DM. In addition, the interactive IM-MOEA algorithm showed in this example that it can follow the changes in the DM's preferences.

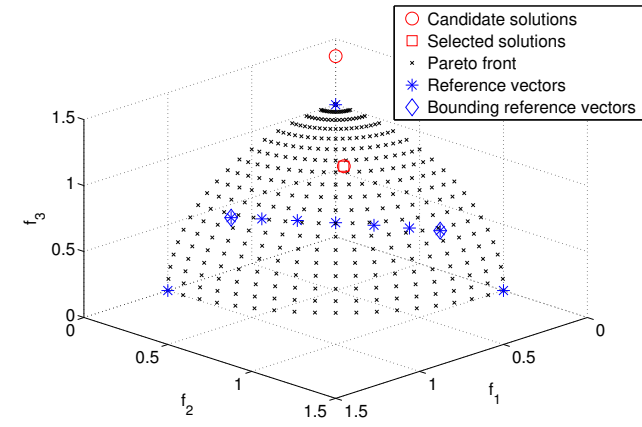
Table 1 Candidate solutions shown to DM during four interactions for problems F2 and F4

Interaction	Solution number	Candidate solutions (problem F2) (f_1, f_2)	Candidate solutions (problem F4) (f_1, f_2, f_3)
1	1	(0.000, 1.000)	(0.146, 41.3, 0.012)
	2	(0.122, 0.988)	(0.000, 6.81, 27.8)
	3	(0.267, 0.932)	(73.9, 0.022, 0.002)
	4	(0.415, 0.830)	(4.95, 6.27, 9.39)
	5		(26.8, 0.000, 33.2)
2	1	(0.507, 0.747)	(0.553, 0.602, 1.01)
	2	(0.475, 0.788)	(0.000, 0.000, 1.37)
	3	(0.354, 0.878)	(0.001, 12.7, 0.000)
	4	(0.312, 0.907)	(19.0, 1.56, 0.000)
	5	(0.000, 1.000)	(5.33, 1.04, 5.90)
3	1	(0.000, 1.000)	(0.000, 0.000, 1.03)
	2	(0.482, 0.770)	(0.297, 0.747, 0.718)
	3	(0.471, 0.780)	(0.015, 1.54, 0.212)
	4	(0.502, 0.751)	(0.654, 0.353, 0.750)
	5	(0.492, 0.761)	(4.79, 0.000, 1.05)
4	1	(0.000, 1.000)	(0.413, 0.889, 0.282)
	2	(0.361, 0.873)	(1.67, 0.000, 0.101)
	3	(0.224, 0.953)	(0.002, 1.03, 0.000)
	4	(0.149, 0.980)	(0.000, 0.000, 1.02)
	5	(0.471, 0.780)	(0.649, 0.603, 0.569)
5	1		(0.000, 0.000, 1.06)
	2		(0.587, 0.693, 0.495)
	3		(0.656, 0.558, 0.553)
	4		(1.02, 0.000, 0.002)
	5		(0.049, 1.00, 0.008)

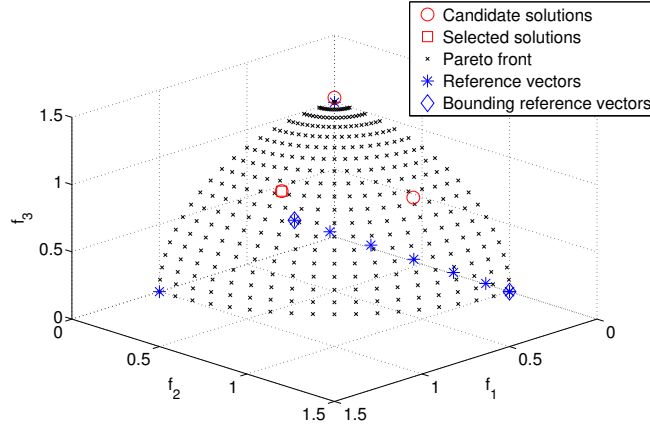
3.2 Example with a three objective problem

To illustrate the behavior of the interactive IM-MOEA algorithm in three objective problems, we use the problem F4 from [3]. In this case, the DM wants to interact five times ($N_{ia} = 5$) and, again, he wants to see five solutions at each interaction ($N_{DM} = 5$).

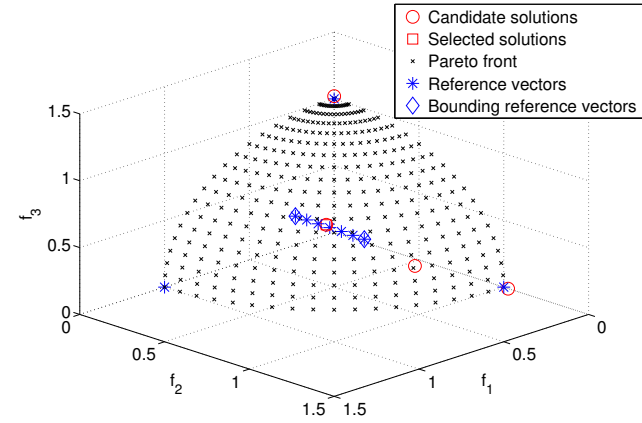
Initially, his preferences are to get as well balanced solution between the three objectives as possible. In the first interaction, the candidate solutions shown to him are presented in Table 1. None of the candidate solutions are acceptable for the DM (they are still very far from the Pareto front) and, thus, the DM does not select any of the solutions. This means that only one of the reference vectors is updated according to step 1 in Algorithm 2.



Interaction 2



Interaction 3



Interaction 4

Fig. 3 Interactions 2-4 with the DM for Problem F4

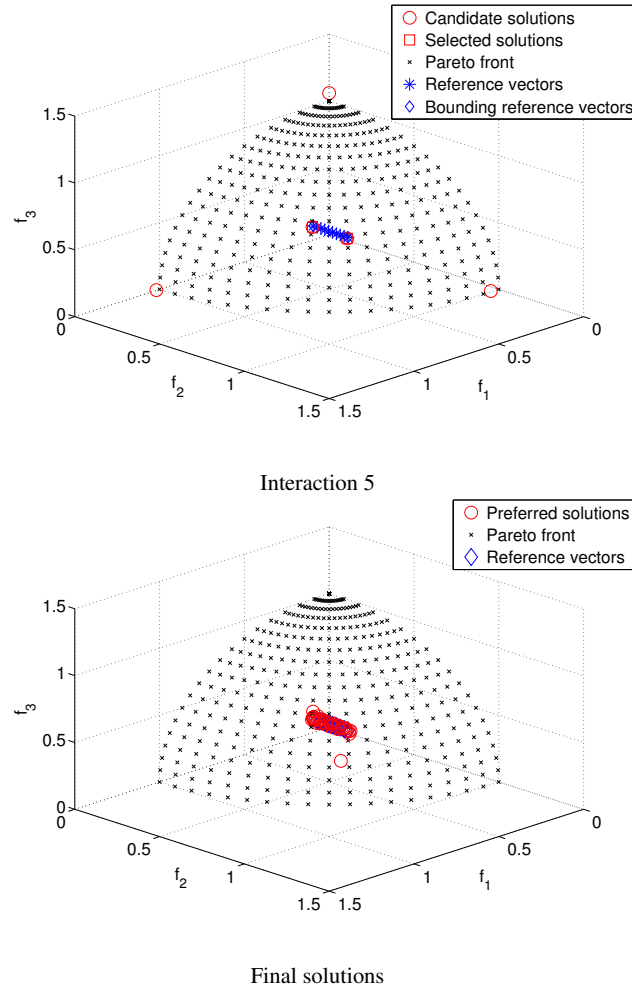


Fig. 4 Interaction 5 with the DM and final solutions for Problem F4

The candidate solutions for the second interaction are shown in Table 1. Although many solutions are still not acceptable, the candidate solution number 1 seems to be best of them for the DM and, therefore, he selects that one. Figures 3 and 4 show the candidate solutions, solutions selected by the DM and the updated set of the reference vectors where the bounding reference vectors are highlighted. In addition, the real Pareto front is also illustrated here to ease the visualization of the performance. For practical problems, the Pareto front is not known and, therefore, can not be visualized to the DM. The resulting candidate solutions for the third interaction are shown in Table 1.

Since most of the candidate solutions start to be in the ranges acceptable for the DM, next he wants to aim at finding a good balance between the objectives. There-

fore, he selected candidate solutions 3 and 4 and the Figure 3 shows the candidate solutions, solutions selected by the DM and the updated set of the reference vectors where the bounding reference vectors are highlighted. As can be seen in Figure 3, he tries to aim towards the middle region of the Pareto front. The resulting candidate solutions are shown in Table 1.

Based on the candidate solutions obtained, number 5 seems to represent a good compromise between the objectives so that is selected by the DM next. Again, the resulting data is shown in Figure 4 and the resulting candidate solutions for the last interaction can be seen in Table 1.

Among the resulting candidate solutions, three (1, 4 and 5) represent solutions where one of the objectives has not so good value and the other two (2 and 3) represent good balance between all the objectives. Therefore, the DM chose 2 and 3 (see Figure 4 for illustration). The final solutions obtained are shown in Figure 4 and it can be seen that they approximate balanced compromises between the objectives.

4 Conclusions

In this paper we have proposed a novel interactive IM-MOEA algorithm that progressively considers the preference information of the DM and finally generates a set of solutions desirable to the DM. This algorithm strives to minimize the cognitive burden on the DM by providing the DM flexibility to decide on the number of solutions (s)he wishes to investigate and how many times (s)he wishes to interact. The algorithm presented considers a simple approach of generating reference vectors in the regions preferable to the DM, thereby generating preferred solutions. The results indicate the ease with which the DM was able to direct his search using the interactive IM-MOEA algorithm and find her/his preferred solution.

Our future research topics include enhancing reference vector adaptation for problems having more than two objectives and adding constraint handling to the proposed algorithm. In addition, testing the algorithm with real-world problems having larger number of objectives is required, for example, the General Aviation Aircraft problem [13].

Acknowledgements This research was funded by Tekes, the Finnish Funding Agency for Innovation (DeCoMo project). The authors greatly appreciate discussions with Prof. Yaochu Jin and Prof. Kaisa Miettinen related to the ideas presented in this paper as well as MTech Tinkle Chugh for his insight in the IM-MOEA algorithm.

References

1. Branke, J.: Consideration of partial user preferences in evolutionary multiobjective optimization. In: J. Branke, K. Deb, K. Miettinen, R. Slowinski (eds.) *Multiobjective Optimization: Interactive and Evolutionary Approaches*, pp. 157–178. Springer-Verlag (2008)

2. Cheng, R., Jin, Y., Narukawa, K.: Adaptive reference vector generation for inverse model based evolutionary multiobjective optimization with degenerate and disconnected pareto fronts. In: A. Gaspar-Cunha, C. Henggeler Antunes, C. Coello Coello (eds.) *Evolutionary Multi-Criterion Optimization*, pp. 127–140. Springer (2015)
3. Cheng, R., Jin, Y., Narukawa, K., Sendhoff, B.: A multiobjective evolutionary algorithm using gaussian process based inverse modeling. *IEEE Transactions on Evolutionary Computation* **19**(6), 838–856 (2015)
4. Deb, K.: *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Ltd., Chichester (2001)
5. Deb, K., Chaudhuri, S.: I-EMO: An interactive evolutionary multi-objective optimization tool. In: S.K. Pal, S. Bandyopadhyay, S. Biswas (eds.) *Pattern Recognition and Machine Intelligence*, pp. 690–695. Springer, Berlin Heidelberg (2005)
6. Deb, K., Sinha, A., Korhonen, P., Wallenius, J.: An interactive evolutionary multiobjective optimization method based on progressively approximated value functions. *Evolutionary Computation, IEEE Transactions on* **14**(5), 723–739 (2010)
7. Jain, A., Murty, M., Flynn, P.: Data clustering: A review. *ACM Computing Surveys* **31**, 264–323 (1999)
8. Jaskiewicz, A., Branke, J.: Interactive multiobjective evolutionary algorithms. In: J. Branke, K. Deb, K. Miettinen, R. Slowinski (eds.) *Multiobjective Optimization: Interactive and Evolutionary Approaches*, pp. 179–194. Springer, Berlin (2008)
9. Miettinen, K.: Survey of methods to visualize alternatives in multiple criteria decision making problems. *OR Spectrum* **36**(1), 3–37 (2014)
10. Miettinen, K., Ruiz, F., Wierzbicki, A.P.: Introduction to multiobjective optimization: Interactive approaches. In: J. Branke, K. Deb, K. Miettinen, R. Slowinski (eds.) *Multiobjective Optimization: Interactive and Evolutionary Approaches*, pp. 27–57. Springer, Berlin (2008)
11. Purshouse, R., Deb, K., Mansor, M., Mostaghim, S., Wang, R.: A review of hybrid evolutionary multiple criteria decision making methods. In: *Evolutionary Computation (CEC), 2014 IEEE Congress on*, pp. 1147–1154 (2014)
12. Thiele, L., Miettinen, K., Korhonen, P., Molina, J.: A preference-based evolutionary algorithm for multi-objective optimization. *Evolutionary Computation* **17**(3), 411–436 (2009)
13. Woodruff, M.J., Reed, P.M., Simpson, T.W.: Many objective visual analytics: rethinking the design of complex engineered systems. *Structural and Multidisciplinary Optimization* **48**(1), 201–219 (2013)